It’s lightweight, easy to use, and allows for the computation of many different kinds of correlations, such as **partial** correlations, **Bayesian** correlations, **multilevel** correlations, **polychoric** correlations, **biweight**, **percentage bend** or **Sheperd’s Pi** correlations (types of **robust** correlation), **distance** correlation (a type of **non-linear** correlation) and more, also allowing for combinations between them (for instance, *Bayesian partial multilevel correlation*).

You can install and load the package as follows:

install.packages("correlation")

library(correlation)

**Examples**

The main function is [correlation()](https://easystats.github.io/correlation/reference/correlation.html), which builds on top of [cor\_test()](https://easystats.github.io/correlation/reference/cor_test.html) and comes with a number of possible options.

**Correlation details and matrix**

cor <- correlation(iris)

cor

## Parameter1 | Parameter2 | r | t | df | p | 95% CI | Method | n\_Obs

## ---------------------------------------------------------------------------------------------

## Sepal.Length | Sepal.Width | -0.12 | -1.44 | 148 | 0.152 | [-0.27, 0.04] | Pearson | 150

## Sepal.Length | Petal.Length | 0.87 | 21.65 | 148 | < .001 | [ 0.83, 0.91] | Pearson | 150

## Sepal.Length | Petal.Width | 0.82 | 17.30 | 148 | < .001 | [ 0.76, 0.86] | Pearson | 150

## Sepal.Width | Petal.Length | -0.43 | -5.77 | 148 | < .001 | [-0.55, -0.29] | Pearson | 150

## Sepal.Width | Petal.Width | -0.37 | -4.79 | 148 | < .001 | [-0.50, -0.22] | Pearson | 150

## Petal.Length | Petal.Width | 0.96 | 43.39 | 148 | < .001 | [ 0.95, 0.97] | Pearson | 150

The output is not a square matrix, but a **(tidy) dataframe with all correlations tests per row**. One can also obtain a **matrix** using:

summary(cor)

## Parameter | Petal.Width | Petal.Length | Sepal.Width

## -------------------------------------------------------

## Sepal.Length | 0.82\*\*\* | 0.87\*\*\* | -0.12

## Sepal.Width | -0.37\*\*\* | -0.43\*\*\* |

## Petal.Length | 0.96\*\*\* | |

Note that one can also obtain the full, **square** and redundant matrix using:

as.table(cor)

## Parameter | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width

## ----------------------------------------------------------------------

## Sepal.Length | 1.00\*\*\* | -0.12 | 0.87\*\*\* | 0.82\*\*\*

## Sepal.Width | -0.12 | 1.00\*\*\* | -0.43\*\*\* | -0.37\*\*\*

## Petal.Length | 0.87\*\*\* | -0.43\*\*\* | 1.00\*\*\* | 0.96\*\*\*

## Petal.Width | 0.82\*\*\* | -0.37\*\*\* | 0.96\*\*\* | 1.00\*\*\*

**Grouped dataframes**

The function also supports **stratified correlations**, all within the *tidyverse* workflow!

library(dplyr)

iris %>%

select(Species, Petal.Width, Sepal.Length, Sepal.Width) %>%

group\_by(Species) %>%

correlation()

## Group | Parameter1 | Parameter2 | r | t | df | p | 95% CI | Method | n\_Obs

## ------------------------------------------------------------------------------------------------------

## setosa | Petal.Width | Sepal.Length | 0.28 | 2.01 | 48 | 0.101 | [ 0.00, 0.52] | Pearson | 50

## setosa | Petal.Width | Sepal.Width | 0.23 | 1.66 | 48 | 0.104 | [-0.05, 0.48] | Pearson | 50

## setosa | Sepal.Length | Sepal.Width | 0.74 | 7.68 | 48 | < .001 | [ 0.59, 0.85] | Pearson | 50

## versicolor | Petal.Width | Sepal.Length | 0.55 | 4.52 | 48 | < .001 | [ 0.32, 0.72] | Pearson | 50

## versicolor | Petal.Width | Sepal.Width | 0.66 | 6.15 | 48 | < .001 | [ 0.47, 0.80] | Pearson | 50

## versicolor | Sepal.Length | Sepal.Width | 0.53 | 4.28 | 48 | < .001 | [ 0.29, 0.70] | Pearson | 50

## virginica | Petal.Width | Sepal.Length | 0.28 | 2.03 | 48 | 0.048 | [ 0.00, 0.52] | Pearson | 50

## virginica | Petal.Width | Sepal.Width | 0.54 | 4.42 | 48 | < .001 | [ 0.31, 0.71] | Pearson | 50

## virginica | Sepal.Length | Sepal.Width | 0.46 | 3.56 | 48 | 0.002 | [ 0.20, 0.65] | Pearson | 50

**Bayesian Correlations**

It is very easy to switch to a **Bayesian framework**.

correlation(iris, bayesian=TRUE)

## Parameter1 | Parameter2 | rho | 89% CI | pd | % in ROPE | BF | Prior | n\_Obs

## --------------------------------------------------------------------------------------------------------------

## Sepal.Length | Sepal.Width | -0.12 | [-0.24, 0.01] | 91.60% | 44.45% | 0.51 | Cauchy (0 +- 0.33) | 150

## Sepal.Length | Petal.Length | 0.86 | [ 0.82, 0.89] | 100% | 0% | > 999 | Cauchy (0 +- 0.33) | 150

## Sepal.Length | Petal.Width | 0.81 | [ 0.76, 0.85] | 100% | 0% | > 999 | Cauchy (0 +- 0.33) | 150

## Sepal.Width | Petal.Length | -0.41 | [-0.52, -0.31] | 100% | 0% | > 999 | Cauchy (0 +- 0.33) | 150

## Sepal.Width | Petal.Width | -0.35 | [-0.47, -0.25] | 100% | 0.22% | > 999 | Cauchy (0 +- 0.33) | 150

## Petal.Length | Petal.Width | 0.96 | [ 0.95, 0.97] | 100% | 0% | > 999 | Cauchy (0 +- 0.33) | 150

**Tetrachoric, Polychoric, Biserial, Biweight…**

The correlation package also supports different types of methods, which can deal with correlations **between factors**!

correlation(iris, include\_factors = TRUE, method = "auto")

## Parameter1 | Parameter2 | r | t | df | p | 95% CI | Method | n\_Obs

## --------------------------------------------------------------------------------------------------------------

## Sepal.Length | Sepal.Width | -0.12 | -1.44 | 148 | 0.304 | [-0.27, 0.04] | Pearson | 150

## Sepal.Length | Petal.Length | 0.87 | 21.65 | 148 | < .001 | [ 0.83, 0.91] | Pearson | 150

## Sepal.Length | Petal.Width | 0.82 | 17.30 | 148 | < .001 | [ 0.76, 0.86] | Pearson | 150

## Sepal.Length | Species.setosa | -0.93 | -29.97 | 148 | < .001 | [-0.95, -0.90] | Biserial | 150

## Sepal.Length | Species.versicolor | 0.10 | 1.25 | 148 | 0.304 | [-0.06, 0.26] | Biserial | 150

## Sepal.Length | Species.virginica | 0.82 | 17.66 | 148 | < .001 | [ 0.77, 0.87] | Biserial | 150

## Sepal.Width | Petal.Length | -0.43 | -5.77 | 148 | < .001 | [-0.55, -0.29] | Pearson | 150

## Sepal.Width | Petal.Width | -0.37 | -4.79 | 148 | < .001 | [-0.50, -0.22] | Pearson | 150

## Sepal.Width | Species.setosa | 0.78 | 15.09 | 148 | < .001 | [ 0.71, 0.84] | Biserial | 150

## Sepal.Width | Species.versicolor | -0.60 | -9.20 | 148 | < .001 | [-0.70, -0.49] | Biserial | 150

## Sepal.Width | Species.virginica | -0.18 | -2.16 | 148 | 0.130 | [-0.33, -0.02] | Biserial | 150

## Petal.Length | Petal.Width | 0.96 | 43.39 | 148 | < .001 | [ 0.95, 0.97] | Pearson | 150

## Petal.Length | Species.setosa | -1.00 | -Inf | 148 | < .001 | [-1.00, -1.00] | Biserial | 150

## Petal.Length | Species.versicolor | 0.26 | 3.27 | 148 | 0.007 | [ 0.10, 0.40] | Biserial | 150

## Petal.Length | Species.virginica | 0.93 | 31.09 | 148 | < .001 | [ 0.91, 0.95] | Biserial | 150

## Petal.Width | Species.setosa | -1.00 | -Inf | 148 | < .001 | [-1.00, -1.00] | Biserial | 150

## Petal.Width | Species.versicolor | 0.15 | 1.87 | 148 | 0.191 | [-0.01, 0.31] | Biserial | 150

## Petal.Width | Species.virginica | 0.99 | 112.56 | 148 | < .001 | [ 0.99, 1.00] | Biserial | 150

## Species.setosa | Species.versicolor | -0.88 | -22.35 | 148 | < .001 | [-0.91, -0.84] | Tetrachoric | 150

## Species.setosa | Species.virginica | -0.88 | -22.35 | 148 | < .001 | [-0.91, -0.84] | Tetrachoric | 150

## Species.versicolor | Species.virginica | -0.88 | -22.35 | 148 | < .001 | [-0.91, -0.84] | Tetrachoric | 150

**Partial Correlations**

It also supports **partial correlations**:

iris %>%

correlation(partial = TRUE) %>%

summary()

## Parameter | Petal.Width | Petal.Length | Sepal.Width

## -------------------------------------------------------

## Sepal.Length | -0.34\*\*\* | 0.72\*\*\* | 0.63\*\*\*

## Sepal.Width | 0.35\*\*\* | -0.62\*\*\* |

## Petal.Length | 0.87\*\*\* | |

**Gaussian Graphical Models (GGMs)**

Such partial correlations can also be represented as **Gaussian graphical models**, an increasingly popular tool in psychology:

library(see) # for plotting

library(ggraph) # needs to be loaded

mtcars %>%

correlation(partial = TRUE) %>%

plot()

**Get Involved**

*easystats* is a new project in active development, looking for contributors and supporters. Thus, do not hesitate to contact us if **you want to get involved**